

# APPENDIX

JC526 U.S. PTO  
09/057786  
04/08/98

## IAMTVTuner::AutoTune

Scans for a precise signal on the channel's frequency.

```
HRESULT AutoTune(  
    long IChannel,  
    long * pIFoundSignal  
);
```

### Parameters

*IChannel*

[in] TV channel number.

*pIFoundSignal*

[out] Value indicating whether the channel's frequency was found; TRUE indicates found, FALSE indicates not found.

### Remarks

TV channels generally map to a unique frequency depending on regional variances. To avoid interference between multiple transmitters that are assigned the same channel when they are in close geographic proximity, small frequency offsets are introduced at each transmitter. In the US, this offset ranges up to +/- 26.25 kilohertz (kHz).

This method handles the channel to frequency conversion and scans for the most precise frequency. These values are stored by calling the "IAMTVTuner::StoreAutoTune"\_method.

## IAMTVTuner::ChannelMinMax

Retrieves the highest and lowest channels available.

```
HRESULT ChannelMinMax(  
    long *IChannelMin,  
    long *IChannelMax  
);
```

### Parameters

*IChannelMin*

[out] Pointer to the lowest channel.

*IChannelMax*

[out] Pointer to the highest channel.

## **IAMTVTuner::get\_AudioFrequency**

Retrieves the currently tuned audio frequency.

```
HRESULT get_AudioFrequency(  
    long *lFreq  
);
```

### **Parameters**

*lFreq*  
[out] Pointer to the current audio frequency.

## **IAMTVTuner::get\_AvailableTVFormats**

Retrieves all analog video TV standards that are supported by the tuner.

```
HRESULT get_AvailableTVFormats(  
    long *lAnalogVideoStandard  
);
```

### **Parameters**

*lAnalogVideoStandard*  
[out] Pointer to the combination of analog video standards supported.

## **IAMTVTuner::get\_Channel**

Retrieves the current TV channel set by the "IAMTVTuner::put\_Channel" method.

```
HRESULT get_Channel (  
    long *plChannel,  
    long *plVideoSubChannel,  
    long *plAudioSubChannel  
);
```

### **Parameters**

*plChannel*  
[out] Pointer to the channel.

*plVideoSubChannel*  
[out] Pointer to a predefined video subchannel value. Specify  
AMTUNER\_SUBCHAN\_NO\_TUNE for no tuning or AMTUNER\_SUBCHAN\_DEFAULT  
for default subchannel.

*plAudioSubChannel*  
[out] Pointer to a predefined audio subchannel value. Specify  
AMTUNER\_SUBCHAN\_NO\_TUNE for no tuning or AMTUNER\_SUBCHAN\_DEFAULT  
for default subchannel.

## **IAMTVTuner::get\_ConnectInput**

Retrieves the hardware tuner input connection.

```
HRESULT get_ConnectInput (  
    long *plIndex  
);
```

### **Parameters**

*plIndex*  
[out] Pointer to the input pin to get the connection for.

## **IAMTVTuner::get\_CountryCode**

Retrieves the country code that establishes the current channel to frequency mapping.

```
HRESULT get_CountryCode (  
    long *plCountryCode  
);
```

### **Parameters**

*plCountryCode*  
[in] Country code currently in use by the tuner filter 110 (Fig. 4).

### **Remarks**

The "IAMTVTuner::put\_CountryCode" method determines which channel to frequency mapping table to use. This establishes the base frequencies for the given country. Use the "IAMTVTuner::AutoTune" method to determine the exact frequencies for specific regions.

## **IAMTVTuner::get\_InputType**

Retrieves the input type set in the "IAMTVTuner::put\_InputType" method.

```
HRESULT get_InputType (  
    long lIndex,  
    TunerInputType * pInputType  
);
```

### **Parameters**

*lIndex*  
[in] Index value that specifies the input pin that will be set.

*pInputType*

[out] Pointer to the "TunerInputType" connection type specified in the "IAMTVTuner::put\_InputType" method; either cable (TunerInputCable) or antenna (TunerInputAntenna).

## **IAMTVTuner::get\_NumInputConnections**

Retrieves the number of TV sources plugged into the tuner filter.

```
HRESULT get_NumInputConnections(  
    long * pNumInputConnections  
);
```

### **Parameters**

*pNumInputConnections*

[out] Number of TV sources plugged into the tuner filter.

## **IAMTVTuner::get\_TuningSpace**

Gets the storage index for regional fine tuning set in the "IAMTVTuner::put\_TuningSpace" method.

```
HRESULT get_TuningSpace(  
    long * pTuningSpace  
);
```

### **Parameters**

*pTuningSpace*

[out] Value specifying the current locale.

### **Remarks**

As TV tuners move into portable systems, you must retain locale-specific mappings of available channels and their actual frequencies. Formulating different *ITuningSpace* values for each locale provides a way of switching the channel to frequency mappings when moving from region to region.

## **IAMTVTuner::get\_TVFormat**

Retrieves the current analog video TV standard in use.

```
HRESULT get_TVFormat(  
    long * pAnalogVideoStandard  
);
```

## Parameters

*pAnalogVideoStandard*

[out] Pointer to the analog video standard currently in use by the tuner filter 110 (Fig. 4).

## IAMTVTuner::get\_VideoFrequency

Retrieves the current video frequency.

```
HRESULT get_VideoFrequency(  
    long *IFreq  
);
```

## Parameters

*IFreq*

[out] Pointer to the video frequency.

## IAMTVTuner::put\_Channel

Sets the TV channel.

```
HRESULT put_Channel(  
    long IChannel,  
    long IVideoSubChannel,  
    long IAudioSubChannel  
);
```

## Parameters

*IChannel*

[in] TV channel number.

*IVideoSubChannel*

Predefined video subchannel value. Specify AMTUNER\_SUBCHAN\_NO\_TUNE for no tuning or AMTUNER\_SUBCHAN\_DEFAULT for default subchannel.

*IAudioSubChannel*

Predefined audio subchannel value. Specify AMTUNER\_SUBCHAN\_NO\_TUNE for no tuning or AMTUNER\_SUBCHAN\_DEFAULT for default subchannel.

## Remarks

This method handles the channel to frequency function call that converts the TV channel to a TV frequency.

## **IAMTVTuner::put\_ConnectInput**

Sets the hardware tuner input connection.

```
HRESULT put_ConnectInput(  
    long IIndex  
);
```

### **Parameters**

*IIndex*  
[in] Index value of the input pin to set connection for.

## **IAMTVTuner::put\_CountryCode**

Sets the country code to establish the frequency to use.

```
HRESULT put_CountryCode(  
    long ICountryCode  
);
```

### **Parameters**

*ICountryCode*  
[in] Value indicating the country code.

### **Remarks**

This method establishes the base frequencies for the given country. Use the "IAMTVTuner::AutoTune" method to determine the exact frequencies for specific regions, unless there are previously cached settings for the new country.

## **IAMTVTuner::put\_InputType**

Sets the tuner input type (cable or antenna).

```
HRESULT put_InputType(  
    long IIndex,  
    TunerInputType InputType  
);
```

### **Parameters**

*IIndex*  
[in] Index value that specifies the input pin to be set.  
*InputType*  
[in] Indicates the connection type, as specified in the TunerInputType data type.

## **IAMTVTuner::put\_TuningSpace**

Sets a storage index for regional channel to frequency mappings.

```
HRESULT put_TuningSpace(  
    long ITuningSpace  
);
```

### **Parameters**

*ITuningSpace*  
[in] Value indicating the current locale.

### **Remarks**

For portable systems, this method retains locale-specific mappings of available channels and their actual frequencies. Formulating different *ITuningSpace* values for each locale provides a way of switching the channel to frequency mappings when moving from region to region.

## **IAMTVTuner::StoreAutoTune**

Saves the fine-tuning information for all channels.

```
HRESULT StoreAutoTune( );
```

### **Remarks**

Override the channel to frequency information stored by this method by setting a new country code in the "IAMTVTuner::put\_CountryCode" method.

